一种加强 SSD 小目标检测能力的 Atrous 滤波器设计

温捷文，战荫伟，李楚宏，卢剑彪

(广东工业大学 计算机学院，广州 510006)

摘　要：针对实时目标检测 SSD（single shot multiBox detector）算法对小目标检测能力偏差的问题，提出了一种提高特征图分辨率的 Atrous 滤波器设计策略。改进算法在 SSD 网络结构的基础上，把第三、四层卷积层产生的特征图经过规范化后连接在一起，然后通过 Atrous 卷积运算提高这些特征图分辨率。这些特征图共同提供小目标的所需的特征。另外该 SSD 改进算法还加入 SeLU（scaled exponential linear units）激活函数，并在数据预处理阶段设计了一套数据增广方法。实验表明，该改进算法框架相对于原 SSD 算法框架具有更高的检测精度、更优良的鲁棒性，以及在小目标检测上效果明显。

关键词：SSD；目标检测；Atrous

中图分类号：TP3

Design of Atrous filter to strengthen small object detection capability of SSD

Wen Jiewen, Zhan Yinwei, Li Chuhong, Lu Jianbiao

(*School of Computer Science & Technology Guangdong University of Technology*, *Guangzhou* 510006, *China*)

Abstract: In order to overcome the shortcomings that SSD (Single Shot MultiBox Detector) can not detect small objects well, this paper proposed an Atrous filter design strategy, which can strengthen the resolution of feature maps. The improved algorithm concatenated the feature maps that generated by the third and fourth convolution layer after normalization, and then improves the resolution of these feature maps by Atrous computed. The concatenated feature maps provide the required features for small objects. In addition, the SSD improved algorithm also add SeLU (Scaled Exponential Linear Units) activation function and designed a data augmented methods in the data preprocessing phase. The experimental results shows that the proposal algorithm has higher detection accuracy and better robustness than the original SSD algorithm. Furthermore, the detection performance obvious better on small target detection.

Key Words: SSD; object detection; Atrous

0　引言

目标检测是近年来计算机视觉领域最热门最具挑战性的研究课题之一。它在现实世界中有非常重要的应用，如自动驾驶、智能监控等。特别地，深度学习这一方法体系的理论研究以及现实应用为计算机视觉提供了有力支持。

目标检测算法分为传统目标检测算法和结合深度学习的目标检测算法。传统目标检测算法在特征提取阶段需要人工干预来获取原始图像输入中与目标相关的特征信息，一方面，这种需要人工干预的特征获取严重地依赖于特征设计人员的先验知识，效率比较低；另一方面，特征提取阶段一些丢失的有用信息的不可复性，使得特征在分类训练时错误率增大。DPM(deformable parts model)[1]为传统目标检测的经典算法框架，采用 HOG(histogram of gradient)[2]特征与 SVM 分类器结合的策略，在一些特定的检测任务上取得非常好的效果。结合深度学习的目标检测算法又分为基于候选区域和基于回归方法两种。基于候选区域的目标检测算法框架的经典文献有[3~9]等，这些算法框架候选框的产生方法有 SS(selective search)[10]、EB(edge boxes)[11]、RPN(region proposal network)[6]。其中，文献[6]所提出的 RPN 算法使用卷积神经网络直接产生候选区域，将一直以来分离的候选区域和卷积神经网络分类融合到了一起；另外，RPN 采用的 anchor 机制能够比较准确地映射出目标边框的坐标位置，也因此在一定程度上减小了目标检测时的定位误差，从而提高检测精度。目前，基于候选区域的这一系列检测算法还是检测领域的研究主流，但是这些算法的检测速度普遍达不到实时要求。为了解决目标检测算法在检测速度上的瓶颈，基于回归方法的检测算法 YOLO[12]和 SSD[13]，以及它们的改进算法 YOLOv2[14]、DSSD[15]相继产生。YOLO 算法框架设计出

作者简介：温捷文（1991-），男，广东梅州人，硕士研究生，主要研究方向为深度学习、目标检测（w_jiewen@qq.com）；战荫伟（1966-），男，吉林长春人，教授，博士，主要研究方向为图像处理、人机交互、虚拟现实、增强现实；李楚宏（1992-），男，广东汕头人，硕士研究生，主要研究方向为行人检测；卢剑彪（1990-），男，江西吉安人，硕士研究生，主要研究方向为行人检测.

神经网络结构 Darknet 模型，以整张图作为网络的输入，把目标检测问题转换成一个回归问题。该算法在网络输出层直接回归目标边框的坐标位置位置和目标所属的类别。在 GPU 支持下，检测速度达到 45 fps。由于损失函数的设计，YOLO 算法分类误差和定位误差都比较大，算法泛化能力也比较弱，YOLOv2 对此在数据输入、网络结构、定位方法等作了比较大的改进。目前 YOLOv2 是检测速度和检测精度综合性能最好的一种算法。

本文研究的 SSD 算法框架有许多优点：首先，在检测速度上，将 Faster R-CNN 卷积层以及全连接层的网络结构转换为全卷积的网络结构，这一改变，使得目标检测速度得到很大的提升；其次，在检测精度上，将 Faster R-CNN 的 RPN 提取候选区域的 anchor 机制变为在各个尺度的特征图上进行，每一个特征图上的像素对应几个 anchor，网络对 anchor 进行训练，同时驱动对特征进行训练，这使得目标检测精度也非常高。

但是研究发现，SSD 算法框架对于小目标的检测能力有限：第一，SSD 是一种基于全卷积网络的检测框架，它用卷积神经网络的不同层检测不同大小的目标。网络模型中，前面的特征图面积大，但上下文语义不够；后面的特征上下文语义丰富，但经过比较多的池化操作，特征图非常小。在 SSD 使用的 VGG-16 网络结构，一个 32×32 大小的目标，经过 conv5_3 层后对应的特征图大小仅为 2×2，位置信息有较大的损失。所以，要检测小目标，既需要一张足够大的特征图来提供更加精细的特征和做更加密集的采样，同时也需要丰富的语义信息与背景区分。第二，SSD 算法框架的网络模型是全卷积网络，可以接受任意大小的图像输入，而不用要求所有的训练图像和测试图像具有同样的尺寸。为了实时速度上的考虑，笔者固定了图片输入尺寸，这对大尺寸图片的小目标检测影响比较大。

为了解决这两个问题，本文提出三种解决办法：a)在比较浅层但是具有高分辨率的卷积层中进行特征采样，同时，采用 Atrous[16]滤波器增加某些特征图的尺寸，提高这些特征图的分辨率，从而为算法提供更有效的特征；b)采用 SeLU 激活函数，使模型训练更具鲁棒性；c)同时也设计了一套与原文相异的数据增广的规则，这套规则使网络模型可以输入不同尺寸大小的图像。

本文总结目前学术界在加强特征提取方面的几个研究工作；介绍 Atrous 滤波器，并给出针对 SSD 算法框架的 Atrous 滤波器设计；针对 ReLU(rectified linear unit)和 SeLU 激活函数进行比较分析；给出一组新的数据增广规则；进行实验分析。

1　相关工作

目标检测或分类，需要算法提供有效的特征。因此，大多数深度学习目标检测算法框架或者分类识别模型为了得到更有效的特征，一般从两个方面考虑：一是多尺度策略；二是采用更深更有效的神经网络结构模型。

在多尺度策略方面有两种处理方法：第一，结合卷积神经网络模型多个卷积层产生的特征图做预测。ION[17]使用 L^2 规范化，在网络结构不同层进行特征采样，进而生成目标候选区域。HyperNet[18]也采用相似的方法来进行特征采样和候选区域生成。不同层的特征图包含输入图像不同层次水平的特征信息，这些信息有利于目标的定位和分类。但是，这种做法也增加了模型的计算量，降低了目标检测速度。第二，不同的卷积层可以设计出不同大小的感受野，预测大目标可选用尺寸大的感受野，相应地，预测小目标选用尺寸小的感受野。所以，可在神经网络模型不同层上采用不同的尺度预测目标。MS-CNN[19]在一个卷积神经网络的多个层应用反卷积的方法增加特征图的分辨率，然后在这些层进行候选区域的学习以及特征采样。为了更好地检测小目标，这些方法使用了小尺寸的浅层感受野和密集特征图的一些目标信息，如上下文语义信息。

在深度神经网络结构方面[26]，基于深度学习的目标检测框架为了达到更好的分类识别和目标检测精度，采用了更深的网络结构。目前比较经典常用的网络结构包括 AlexNet[20]、VGG[21]、GoogleNet[22]、ResNet[23]等。SSD 算法框架采用 VGG-16 模型作为基础网络结构，使用该模型的前五层，然后将第六(fc6)和第七(fc7)层的全连接层转换成两个卷积层，再另外增加了三个卷积层以及一个平均池化(average pooling)层。作为对 SSD 的改进算法，DSSD 除了使用反卷积增加特征图的分辨率、加强上下文语义学习之外，还采用了结构层次更深的、分类精度更好的 ResNet-101 网络模型。DSSD 检测精度高于 SSD，但同时 DSSD 也失去了检测速度方面的实时性。

2　Atrous 滤波器设计

根据文献[16]，Atrous 滤波器最初在小波变换中进行图像处理。Atrous 可以在任意一层的任一种分辨率下计算卷积激励值。首先考虑一维信号的 Atrous 卷积计算：

$$y[i]=\sum_{k=1}^{K}x[i+r\cdot k]w[k]$$

其中：$x[i]$是一维信号输入；$y[i]$是经过运算的信号输出；$w[k]$为滤波器；K 为该滤波器的长度；速率(rate)参数 r 相当于输入信号采样时的步长。图 1 说明了一维信号在做 Atrous 卷积运算时的具体过程。

图 1(a)是低分辨率输入时稀疏特征提取的标准卷积过程，而(b)是 Atrous 卷积过程。相对于图 1(a)可知(b)延拓更大(pad=2)，设置 rate 为 2，是在输入特征图的矩阵中每个值之间插入 0，然后再进行卷积运算。最后，3 个输入信号经过延拓、插值以及卷积运算，输出 5 个激励值。特征图尺寸由此增大。

在二维的输入图像也有类似操作。图 2 是稀疏特征提取和稠密特征提取过程。

在图 2 中，上一行为稀疏特征提取。给定一张图像，假设有一个下采样操作会降低图像的分辨率，此时设定下采样因子 stride 值为 2，然后与核大小为 7 的高斯核进行卷积运算。此时得到的特征图大小只有原图像的 1/4。下一行为稠密特征提取。

本文把滤波器按 stride 值为 1 进行上采样，同时按 rate 值为 2 进行插值。尽管本文算法把滤波器尺寸放大了，但是本文仍然只考虑滤波器的非零值，因此，在每一个位置，滤波器的参数和操作保持不变。这种 Atrous 卷积运算使得本文算法对低分辨率图像的特征响应能够可控。

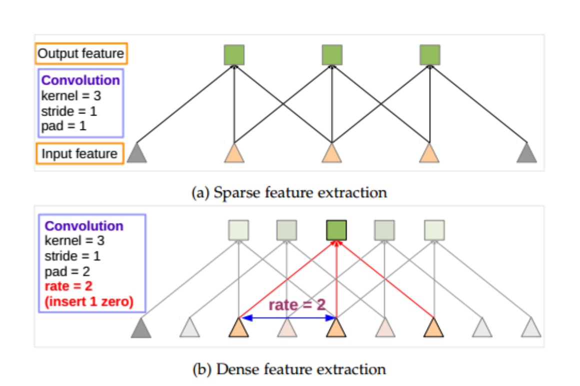


图 1　一维信号特征提取

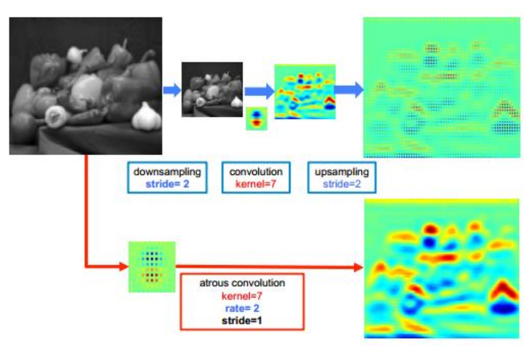


图 2　二维图像特征提取

根据以上描述，如果滤波器大小为 $k\times k$，那么在 rate 值为 r 时，需要在滤波器中插入 r-1 个零值。滤波器尺寸将扩大为

$$k_e = k + (k-1)(r-1)$$

图 3 所示为 SSD 算法框架的网络结构。该框架采用 VGG-16 基础网络结构，使用 VGG-16 的前五层，然后将 VGG-16 网络结构的 fc6 和 fc7 层转换成两个卷积层，再格外增加了四个卷积层，同时移除了所有的 Dropout 层以及 fc8 层。不同层次的特征图分别用于目标边框的偏移以及不同类别得分的预测，最后通过 NMS(non maximum suppression)得到最终的检测结果。

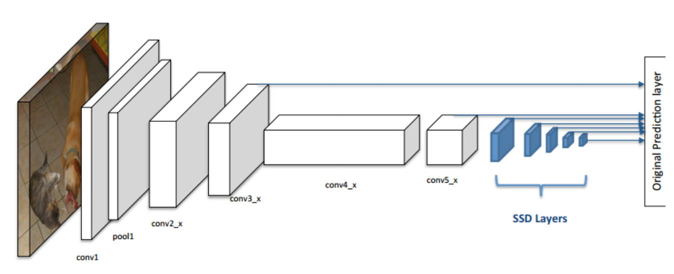


图 3　SSD 网络结构

SSD 中最小尺度的物体检测主要是由 conv4_3 层产生的特征图决定的。该层在整个网络结构中属于比较靠前的位置。本文在 SSD 算法框架的基础上，结合 Atrous 滤波器作用原理，作出如下改进：

a)同时在 VGG-16 基础网络的 conv3_3 和 conv4_3 两层提取特征，此时网络结构的步长设置，在 conv3_3 层 stride 为 4，而在 conv4_3 层 stride 为 8。然后把两个卷积层产生的特征图经过归一化后连接在一起。

b)使用 Atrous 滤波器提高分辨率。本文把网络结构 pool3 层的步长 stride 由 2 设置为 1，然后把 conv4 层的所有滤波器按 rate 值为 2 进行插值扩大，由此提高特征图分辨率。

本文的这两个做法，首先考虑到加强底层的特征信息来源，因此本文把 conv3_3 和 conv4_3 两层产生的特征图经过归一化后连接在一起；然后在 conv4_3 层进行 Atrous 卷积运算加强特征图分辨率。

3　SeLU 激活函数

在 SSD 算法的网络结构中使用的是 ReLU 激活函数。ReLU 激活函数在模型反向传播过程中降低了梯度弥散出现的可能性，神经网络前几层的参数也可以很快地更新；同时正向传播过程中，ReLU 激活函数仅需要设置阈值，这种简单的处理方式加快了正向传播的计算速度。但是训练神经网络模型时，如果使用 ReLU 激活函数，则非常容易导致训练中断。一个大的梯度经过一个 ReLU 神经元，更新过参数之后，这个神经元就不会对任何数据产生激励。所以，使用 ReLU 激活函数，需要设置一个比较小的合适的学习率。

2017 年，文献[24]介绍了一种新的激活函数 SeLU(scaled exponential linear units)。该激活函数定义为

$$f(x) = \lambda \begin{cases} x, & x > 0 \\ \alpha e^x - \alpha, & x \le 0 \end{cases}$$

SeLU 引入了自归一化的属性，使神经元激励值可以自动地收敛到零均值和单位方差。相对于批归一化要求精确的归一化，SeLU 激励值逼近于零均值和单位方差，并且即使是存在噪声和扰动的情况下，通过许多层的前向传播后还是将收敛到零均值和单位方差。这种收敛属性允许：a)训练许多层的深度神经网络；b)采用强正则化；c)令学习更具鲁棒性。此外，对于不逼近单位方差的激励值，其方差存在上确界和下确界，因此梯度消失和梯度爆炸不可能出现。

归一化技术在深度神经网络训练时通常会受到随机梯度下降（SGD）、随机正则化（如 dropout）等参数所扰动。而本文旨在对神经元激励进行自动地转移（shift）和重缩放 (rescale)，在没有明确的归一化的情况下去实现零均值和单位方差。

因此基于上面 SeLU 激活函数的一些优秀特性，本文考虑把 SSD 算法网络结构中 ReLU 激活函数替换为 SeLU 激活函数。本文加入 SeLU 激活函数的目的主要是利用这些特性增加网络结构的鲁棒性。

4　图像预处理

为了让模型对不同大小和形状的目标的检测更具鲁棒性，SSD 采取一定规则对输入图像进行数据增广：

a)使用输入的整幅图像；

b)使采样图像与原始图像比例为 0.1、0.3、0.5、0.7、0.9，用这个采样比例在原始输入图像中采样；

c)随机采样图像。

采样图像的尺寸大小与原始图像大小比例在[0.1,1]间，宽高比在[0.5，2]间。图像采样后，当真实图像(ground truth)的目标边框中心在采样图像中时，SSD 保留重叠部分，然后把采样图像重新放大到固定大小。

研究发现，对于小目标检测，这个采样规则可以进行更好的调整。采样的图像中，目标物体基本尺寸与原始图像物体相比，也是[0.1,1]间的比例。实际上 0.1 的比例还是相对比较大。比如 512×512 的输入图像，按比例 0.1 计算，那么采样后为 51×51。

基于上面的分析，本文针对小目标检测，调整目标物体基本尺寸与原始图像物体比例。本文改进算法使用的比例为[1/64，1]，相应地，改进算法把采样图像与原始图像的采样比例设为 1/64、1/32、1/16、1/8、1/4、1/2、1 七种。

本文这样设计的目的是：a)缩小图像采样的比例，当采样图像重新放大到固定大小时，小目标能够更好地凸现出来；b)SSD 原文采样比例有 5 种，本文设为 7 种，并且有 6 种小于 0.5，这增加了小目标区域的敏感性。

5　实验

5.1　实验准备

本文实验硬件配置为 Intel Xeon E5-2620 v2 处理器、NVIDIA GTX 980ti 显卡、64 GB RAM 的服务器，软件环境为 Ubuntu 系统、GCC、cuda、OpenCV、caffe 框架。其中 GPU 加速采用 cuda 编程，OpenCV 主要为了测试时图片显示。

SSD 算法框架的训练和评估主要在 PASCAL VOC 2007 和 PASCAL VOC 2012 两个数据集上进行。PASCAL VOC 是一个视觉对象的分类识别和检测的标准数据集，提供的图片集包括 20 个类别。表 1 列出了 PASCAL VOC 的具体类别。

表 1　PASCAL VOC 类别

目录	类别
人类	人
动物	鸟、猫、牛、狗、马、羊
交通工具	飞机、自行车、船、公共汽车、小轿车、摩托车、火车
室内	瓶子、椅子、餐桌、盆栽植物、沙发、电视

本文实验把 PASCAL VOC 2007 的验证集和测试集，以及 PASCAL VOC 2012 的验证集一起作为训练集，把 PASCAL VOC 2012 的测试集用来测试。

5.2　评价标准

根据性能侧重点不同，目标检测有许多评价指标，如检测精度、检测效率、定位准确性。本文侧重于目标检测精度。本文采用的 mAP(mean average precision)为目标检测精度最重要的评价指标。其计算流程为：

a)计算每一个类别的平均精度：

$$P=\frac{1}{R}\sum_{j=1}^{n}I_j\frac{R_j}{j}$$

其中：R 表示数据集一个类别所有相关的目标对象个数（检测到和未检测到）；n 表示数据集中目标对象的数量；如果第 j 个目标对象相关，I_j 为 1，否则 I_j 为 0；R_j 是前 j 个目标对象中相关目标对象个数。

b)取多个类别平均精度的平均值。

mAP 值介于 0~1 间，值越大说明算法的检测精度越好。

本文使用每秒检测帧数(frames per second, FPS)衡量目标检测速度,以 25 fps 作为实时性考量临界值。

5.3　实现细节

本文的算法框架改动基于原 SSD 算法框架以及 VGG-16 分类模型，分别在 SSD 网络结构的 conv3_3、pool3、conv4 做了处理，以及用 SeLU 激活函数替换了 ReLU 激活函数。代码实现采用 caffe 框架，并参考了 SSD、SeLU 等论文开源代码。

算法框架在 ImageNet 数据集分类和定位任务上进行模型的预训练，然后再微调为检测模型。本文使用随机梯度下降法，设定初始学习率为 0.001，动量(momentum)为 0.9，权重衰减(decay)为 0.005，批大小(batchsize)为 32。不同数据集的学习率改变策略不同，本文卷积网络结构采用随机初始化策略。在训练数据时，增加训练的迭代次数是非常有必要的。本文实验用 0.001 的学习率迭代训练数据 60 000 次，然后再用 0.000 1 的学习率迭代 30 000 次，最后用 0.000 01 的学习率迭代 10 000 次。

训练过程中，算法需要标定分类的正负样本。正负样本由已标注图片(ground truth)的目标边框与预测的目标边框决定。如果两者的 IOU(intersection-over-union)阈值为 0.5，就设定为正样本，否则设定为负样本。

5.4　实验结果与分析

本文研究做了三组对比实验进行测试、验证改进算法效果。5.4.1 节针对小目标检测对比了网络结构几层的特征提取能力；5.4.2 节是各目标检测算法框架训练效果对比；5.4.3 节是各目标检测算法框架在测试数据集 PASCAL VOC 2012 test 检测精度对比。

5.4.1 Atrous 实验对比

算法在网络结构的较底层进行小目标相关的特征提取。在原 SSD 算法中，conv4_3 层提取的特征对小目标的检测敏感。而本文设计中，把 conv3_3 和 conv4_4 两层的特征图经归一化后连接在一起，这些特征图共同为小目标的检测提供特征。本

文也进行了其他层对小目标特征提取能力的实验。表 2 列出了 SSD 各层特征提取最终效果对比。

表 2　SSD 各层特征提取效果对比

特征提取	fps	mAP/%
conv2_2	59	70.0
conv3_3	59	72.4
conv4_3	**59**	**74.3**
conv5_3	59	68.0
conv3_3, conv4_3	59	74.7
conv3_3, conv4_3, conv5_3	59	73.6
Conv3_3,（conv4_3, Atrous）	**50**	**75.5**

在理论上，网络模型比较前面的卷积层能够为目标检测提供丰富的语义信息，但是这种特征提取能力也不是呈线性增长。本文对比了几个卷积层为小目标提供特征的能力。由表 2 可知，单层提取特征能力 conv4_3 层最强，mAP 为 74.3%，这个也是原 SSD 算法框架中的设计。然后是 conv3_3 层，conv5_3 最差。本文同时把几个卷积层产生的特征图连接在一起，可以看出 conv3_3 和 conv4_3 层两层一起为小目标提供特征，可以使算法 mAP 有所提高。在 conv3_3 和 conv4_3 层连接的特征图的基础上，本文采用 Atrous 滤波器增强这些特征图的分辨率，进一步提供更多的特征信息。此时改进算法的 mAP 可以达到 75.5%，为几个对比实验中最优。但是在检测速度的对比看，单层或者多层直接提供特征是不会增加额外计算的，检测速度保持着 59FPS，而本文的算法设计，Atrous 卷积以及插值运算会增加模型的计算量，使检测速度有所下降，但是仍然能够保持 50 fps 的实时检测速度。

5.4.2 各算法框架训练对比

本文算法框架在 PASCAL VOC 2007 和 PASCAL VOC 2012 两个数据集上训练，实验环境等相关说明在 5.1 节。

本文选取目前几个流行的目标检测算法框架作对比，它们包括 YOLO、YOLOv2、SSD、R-CNN Minus R、Fast R-CNN、Faster R-CNN。本文在训练数据集上主要对比目标检测精度 mAP 和目标检测速度。表 3 对比了几个典型算法框架检测效果。

由表 3 中数据可知，实时目标检测算法框架有 YOLO、YOLOv2、SSD、Ours 四种。相对于 SSD300，本文的改进算法 Ours300 在 mAP 上有 1.2%的提高，但是因为加入了 Atrous 处理，增加了算法框架的计算量，所以改进算法在实时性上减少了 9 fps。Ours512 比 Ours300 检测精度提高 0.4%，但是检测速度减少 7 fps。相对于 SSD512，本文算法的实时性得到非常好的保持。

本文改进算法 Ours 的检测精度，比 YOLOv2 288 和 YOLOv2 352 两种图片输入分辨率下的检测精度要高，在 YOLOv2 544 高分辨率输入情况下，改进算法在实时性上有相对优势。相对于本文的改进算法，YOLO、R-CNN Minus R、Fast R-CNN 以及 Faster R-CNN 在目标检测精度和检测速度方面都没有优势。

表 3　目标检测算法对比

目标检测算法	**训练**	**mAP/%**	**fps**
YOLO	2007+2012	63.4	45
SSD300	**2007+2012**	**74.3**	**59**
SSD512	**2007+2012**	**76.8**	**19**
YOLOv2 288	2007+2012	69.0	91
YOLOv2 352	**2007+2012**	**73.7**	**81**
YOLOv2 416	2007+2012	76.8	67
YOLOv2 480	2007+2012	77.8	59
YOLOv2 544	2007+2012	**78.6**	40
R-CNN Minus R[25]	2007	53.5	6
Fast R-CNN	2007+2012	70.0	0.5
Faster R-CNN	2007+2012	73.2	7
Ours300	**2007+2012**	**75.5**	**50**
Ours512	**2007+2012**	**75.9**	**43**

分析表 3 数据可知，影响检测精度的原因有两点：a)图像的输入尺度，一般图像输入分辨率越高，目标检测精度越好，如 YOLOv2 的五个尺度的图像输入，SSD 和 Ours 的两个尺度的图像输入可对比得出该结论；b)本文的改进算法所使用的 Atrous 设计策略也使检测精度有所提高。相应地，影响检测实时性的原因则是：a)图像输入分辨率越高，所需要的计算量也就越大，实时性有所下降；b)本文使用的 Atrous 滤波器设计策略也在一定程度上增加了计算量，实时性下降。

表 4　PASCAL VOC 2012 test 检测结果

Method	mAP	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	persion	plant	sheep	sofa	train	tv
R-CNN	49.6	68.1	63.8	46.1	29.4	27.9	56.6	57.0	65.9	26.5	48.7	39.5	66.2	57.3	65.4	53.2	26.2	54.5	38.1	50.6	51.6
Fast	68.4	82.3	78.4	70.8	52.3	38.7	77.8	71.6	89.3	44.2	73.0	55.0	87.5	80.5	80.8	72.0	35.1	68.3	65.7	80.4	64.2
Faster	70.4	84.9	79.8	74.3	53.9	49.8	77.5	75.9	88.5	45.6	77.1	55.3	86.9	81.7	80.9	79.6	40.0	72.6	60.9	81.2	61.5
YOLO	57.9	77.0	67.2	57.7	38.3	22.7	68.3	55.9	81.4	36.2	60.8	48.5	77.2	72.3	71.3	63.5	28.9	52.2	54.8	73.9	50.8
SSD300	70.3	84.2	76.3	69.6	53.2	40.8	78.5	73.6	88.0	50.5	73.5	61.7	85.8	80.6	81.2	77.5	44.3	73.2	66.7	81.1	65.8
SSD512	73.1	84.9	82.6	74.4	55.8	50.0	80.3	78.9	88.8	53.7	76.8	59.4	87.6	83.7	82.6	81.4	47.2	75.5	65.6	84.3	68.1
Ours300	**71.4**	84.2	77.1	**73.0**	53.4	**46.1**	87.1	75.3	88.0	51.6	73.2	62.2	85.9	81.1	80.2	77.0	**45.2**	73.3	66.0	81.3	66.1
Ours512	73.3	**85.1**	82.3	**74.6**	**56.0**	**51.3**	**87.5**	76.4	88.9	52.2	77.0	62.0	**88.3**	81.5	**84.6**	**81.8**	**48.0**	73.0	**66.1**	83.4	66.7

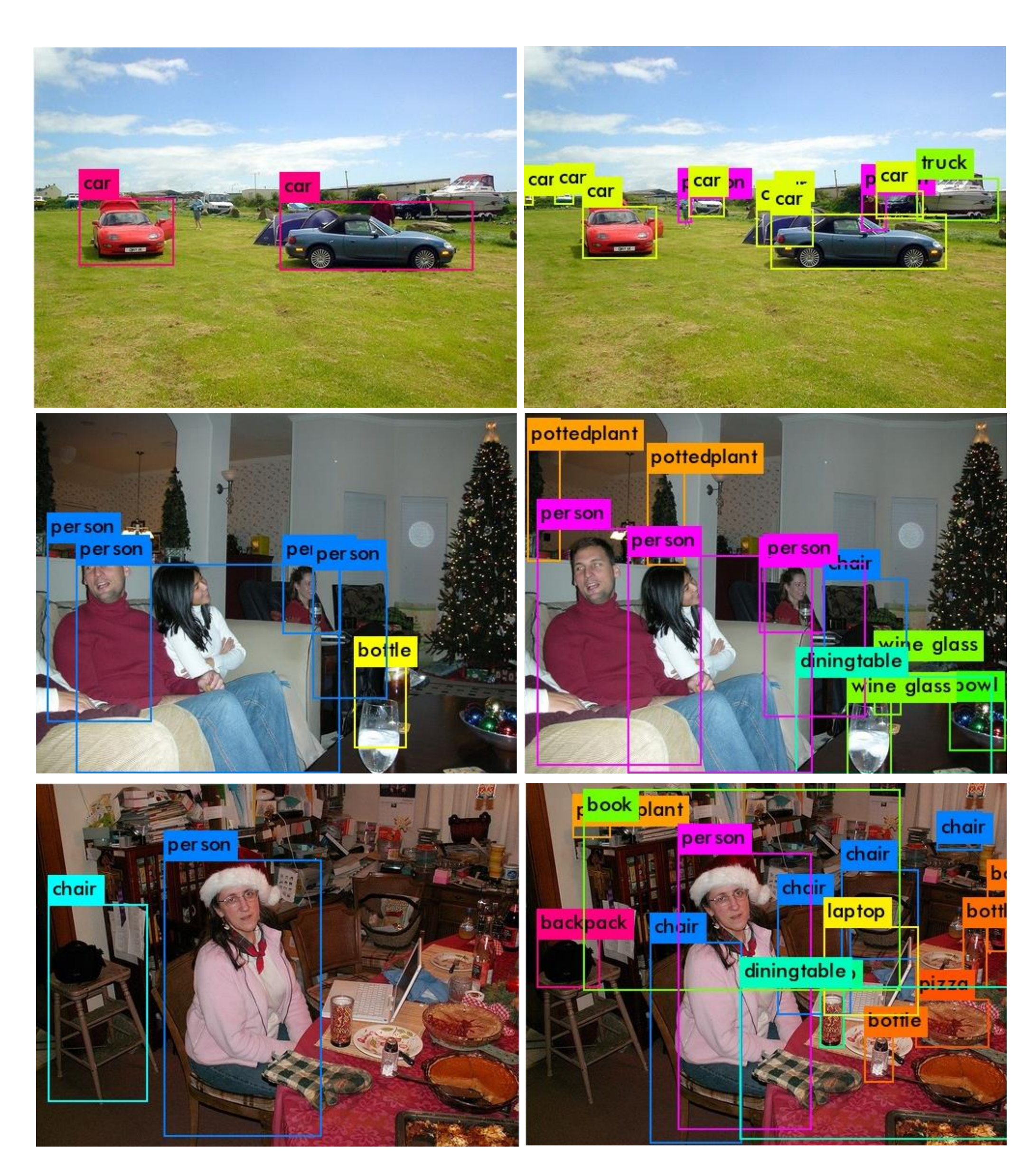


图 4　SSD 检测效果与改进算法检测效果

5.4.3 PASCAL VOC 2012 test 检测对比

本文把各种算法在 PASCAL VOC 2012 test 上进行测试对比。表 4 记录了各种算法在 PASCAL VOC 2012 test 上各个类别的检测精度和平均检测精度 mAP

由表 4 对比可知，本文的改进算法 Ours 不仅在大目标的测试结果中表现优秀，而且几类小目标的检测效果也在几个对比算法中是最好的。在 Ours512 中，大目标如 aero、boat、bus、dog、mbike、person、sofa 等检测精度最高，同时在小目标方面，bird、bottle、plant 等检测精度也最高。

分析表 4 中数据可知，相对于 R-CNN、YOLO、Fast R-CNN，单项检测精度方面 Faster R-CNN、SSD、Ours 都比较高。原因在于 R-CNN 把图像感兴趣区域(region of interest, ROI)分的数量太多，在每一个 ROI 上使用卷积神经网络进行特征提取时，总的计算量十分大，影响了目标检测水平。YOLO 在网络结构中的最后两个全连接层，在回归定位坐标位置方面比较不准确，这直接影响了 YOLO 的定位精度。而 Faster R-CNN、SSD 以及本文的改进算法 Ours，在目标定位方面都采用或者借鉴改进了 RPN，定位误差较小，从而提高了目标检测精度。

5.4.4 改进算法效果

图 4 是原 SSD 算法框架和本文的 SSD 改进算法框架的实验效果对比。由该组图可知，相对于原 SSD，本文的改进算法 Ours 能够检测的目标更多。原 SSD 算法把图片中误判为背景的目标对象，本文改进算法 Ours 能够比较准确地定位并分类。

6　结束语

本文研究了实时目标检测 SSD 算法框架并提出了一种改进算法。该改进算法用 Atrous 滤波器提高特征图分辨率，并为图像预处理阶段设计一套数据增广规则；此外 Ours 在网络结构中用 SeLU 激活函数替换了 ReLU 激活函数。实验结果表明，本文的改进算法 Ours 在小目标检测方面相对于原 SSD 检测算

法表现更为优异。笔者下一步研究工作将在此基础上，结合小波分析等传统方法，尝试对经过 Atrous 滤波器处理后的特征图得出最佳分辨率。

参考文献：

[1] Felzenszwalb P F, Girshick R B, McAllester D, et al. Object detection with discriminatively trained part-based models [J]. IEEE Trans on Pattern Analysis and Machine Intelligence, 2010, 32 (9): 1627-1645.

[2] Dalal N, Triggs B. Histograms of oriented gradients for human detection [C]// Proc of Computer Vision and Pattern Recognition. 2005: 886-893.

[3] Girshick R, Donahue J, Darrell T, et al. Rich feature hierarchies for accurate object detection and semantic segmentation [C]// Proc of IEEE Conference on Computer Vision and Pattern Recognition. 2014: 580-587.

[4] He K, Zhang X, Ren S, et al. Spatial pyramid pooling in deep convolutional networks for visual recognition [J]. IEEE Trans on Pattern Analysis and Machine Intelligence, 2015, 37 (9): 1904-1916.

[5] Girshick R. Fast R-CNN [C]// Proce of IEEE International Conference on Computer Vision. 2015: 1440-1448.

[6] Ren S, He K, Girshick R, et al. Faster R-CNN: towards real-time object detection with region proposal networks [C]// Advances in Neural Information Processing Systems. 2015: 91-99.

[7] Dai J, Li Y, He K, et al. R-FCN: Object detection via region-based fully convolutional networks [C]// Proc of Neural Information Processing Systems. 2016: 379-387.

[8] Lin T Y, Dollár P, Girshick R, et al. Feature pyramid networks for object detection [J]. arXiv preprint arXiv: 1612. 03144, 2016.

[9] He Kaiming, Gkioxari G, Dollár P, et al. Mask R-CNN [J]. arXiv preprint arXiv: 1703. 06870, 2017.

[10] Uijlings J R R, Van De Sande K E A, Gevers T, et al. Selective search for object recognition [J]. International Journal of Computer Vision, 2013, 104 (2): 154-171.

[11] Zitnick C L, Dollár P. Edge boxes: locating object proposals from edges [C]// Proc of European Conference on Computer Vision. [S. l.] : Springer International Publishing, 2014: 391-405.

[12] Redmon J, Divvala S, Girshick R, et al. You only look once: unified, real-time object detection [C]// Proc of IEEE Conference on Computer Vision and Pattern Recognition. 2016: 779-788.

[13] Liu W, Anguelov D, Erhan D, et al. SSD: single shot multibox detector [C]// Proc of European Conference on Computer Vision. Springer International Publishing. 2016: 21-37.

[14] Joseph R, Farhadi A. YOLO9000: better, faster, stronger [J]. arXiv preprint arXiv: 1612. 08242, 2016.

[15] Fu Chengyang, Liu Wei, Ranga A, et al. DSSD: deconvolutional single shot detector [J]. arXiv preprint arXiv: 1701. 06659, 2017.

[16] Chen L C, Papandreou G, Kokkinos I, et al. DeepLab: semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs [J]. arXiv preprint arXiv: 1606. 00915, 2016.

[17] Bell S, Zitnick C L, Bala K, et al. Inside-outside net: detecting objects in context with skip pooling and recurrent neural networks [C]// Proc of IEEE Conference on Computer Vision and Pattern Recognition. 2016: 2874-2883.

[18] Kong Tao, Yao Anbang, Chen Yurong, et al. HyperNet: towards accurate region proposal generation and joint object detection [C]// Proc of IEEE Conference on Computer Vision and Pattern Recognition. 2016: 845-863.

[19] Cai Zhaowei, Fan Quanfu, Rogerio S, et al. A unified multi-scale deep convolutional neural network for fast object detection [C]// Proc of European Conference on Computer Vision. Springer International Publishing. 2016: 354-370.

[20] Krizhevsky A, Sutskever I, Hinton G E. Imagenet classification with deep convolutional neural networks [C]// Advances in Neural Information Processing Systems. 2012: 1097-1105.

[21] Simonyan K, Zisserman A. Very deep convolutional networks for large-scale image recognition [J]. arXiv preprint arXiv: 1409. 1556, 2014.

[22] Szegedy C, Liu W, Jia Y, et al. Going deeper with convolutions [C]// Proc of IEEE Conference on Computer Vision and Pattern Recognition. 2015: 1-9.

[23] He K, Zhang X, Ren S, et al. Deep residual learning for image recognition [C]// Proc of IEEE Conference on Computer Vision and Pattern Recognition. 2016: 770-778.

[24] Klambauer G, Unterthiner T, Mayr A, et al. Self-normalizing neural networks [J]. arXiv preprint arXiv: 1706. 02515, 2017.

[25] Lenc K, Vedaldi A. R-CNN minus r [J]. arXiv preprint arXiv: 1506. 06981, 2015.

[26] 周飞燕，金林鹏，董军. 卷积神经网络研究综述 [J]. 计算机学报, 2017, 40 (6): 1229-1251.